

## System and method for importing and exporting data between an object oriented computing environment and an external computing environment

Patent Number: ☒ US5497491  
Publication date: 1996-03-05  
Inventor(s): MITCHELL HERMAN (US); RICH WILLIAM L (US); SHACKELFORD FLOYD W (US)  
Applicant(s): IBM (US)  
Requested Patent: ☒ JP7175730  
Application Number: US19930009592 19930126  
Priority Number(s): US19930009592 19930126  
IPC Classification: G06F9/44; G06F15/16  
EC Classification: G06F9/44M  
Equivalents: JP2680255B2

---

### Abstract

---

A system and method for importing and exporting data between an object oriented computing environment and an external computing environment encapsulates the validation, conversion, and update control mechanisms for objects in metadata about the objects. Metadata logic operates identically on a variety of internal objects, freeing the object oriented computing environment from the need to duplicate the same control logic for each user interface. During program generation, metadata objects are generated to link an external computing environment data field name, a corresponding object oriented computing environment attribute name and a corresponding external data format for each external data field which is imported and exported. An external class object is defined, including conversion parameters for converting the corresponding object oriented computing environment attribute into the corresponding external data format. External object class metadata objects contain the object oriented computing environment attributes which are used for import and export.

---

Data supplied from the **esp@cenet** database - I2

BEST AVAILABLE COPY

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-175730

(43) 公開日 平成7年(1995)7月14日

(51) Int.Cl. <sup>6</sup>	識別記号	片内整理番号	F I	技術表示箇所
G 0 6 F 13/00	3 5 1 A	7368-5B		
9/44	5 3 0 M	9193-5B		

審査請求 有 請求項の数17 O L (全 21 頁)

(21) 出願番号 特願平5-332725

(22) 出願日 平成5年(1993)12月27日

(31) 優先権主張番号 0 0 9 5 9 2

(32) 優先日 1993年1月26日

(33) 優先権主張国 米国 (US)

(71) 出願人 390009531

インターナショナル・ビジネス・マシーンズ・コーポレーション

INTERNATIONAL BUSINESS MACHINES CORPORATION

アメリカ合衆国10504、ニューヨーク州アーモンク (番地なし)

(72) 発明者 ハーマン・ミッチェル

アメリカ合衆国30038、ジョージア州リスオニア、エバンス・ミル・ロード 3793

(74) 代理人 弁理士 合田 潔 (外2名)

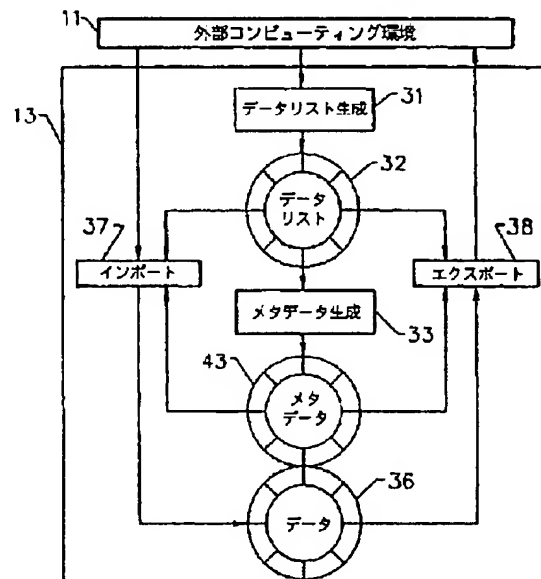
最終頁に続く

(54) 【発明の名称】 オブジェクト指向環境においてデータを転送するためのシステム及び方法

(57) 【要約】

【目的】 他のコンピューティング環境との間でデータをインポート及びエクスポートするためのシステム及び方法を含む、オブジェクト指向コンピューティング環境を提供すること。

【構成】 本発明によればこれらの目的が、オブジェクトに対する妥当性検証、変換及び更新制御機構を、オブジェクトに関する"メタデータ" (metadata) にカプセル化することにより達成される。このカプセル化は、メタデータ論理が様々な内部オブジェクトに対し同様に動作することを可能とし、各ユーザ・インタフェースに対応する同一の制御論理を複製する必要性から、オブジェクト指向コンピューティング環境を解放する。データのインポート及びエクスポートのための変換及び検証パラメータを定義するために、オブジェクトの属性を変更するタグ付け機構を使用することにより、データがインポート及びエクスポートされる形式及びコンテキストが一般的に定義される。



1

## 【特許請求の範囲】

【請求項1】 1つまたは複数のコンピュータ・プラットフォーム上で動作する外部コンピューティング環境と、  
1つまたは複数のコンピューティング・プラットフォーム上で動作するオブジェクト指向コンピューティング環境との間でデータをインポート及びエクスポートするためのシステムであって、

外部コンピューティング環境データ・フィールド名と、  
対応するオブジェクト指向コンピューティング環境属性と、  
上記外部コンピューティング環境における各外部データ・フィールドに対応する外部データ形式とをリストするデータリスト・オブジェクトを生成するデータリスト生成手段と、

対応する上記オブジェクト指向コンピューティング環境属性の1つを含む、上記オブジェクト指向コンピューティング環境内の各オブジェクトに対応するメタデータ・オブジェクトを生成するメタデータ生成手段であって、  
該メタデータ・オブジェクトが、対応するオブジェクト指向コンピューティング環境属性を上記対応する外部データ形式に変換する変換パラメータを含むメタデータ生成手段と、

外部データを上記オブジェクト指向コンピューティング環境にインポートするための上記外部コンピューティング環境からの要求に回答して、上記データリスト・オブジェクトからオブジェクト指向コンピューティング環境属性を識別し、対応するメタデータ・オブジェクトを識別し、該対応するメタデータ・オブジェクト内の変換パラメータを上記外部データに適用し、それにより該外部データをオブジェクト指向データに変換し、該変換されたオブジェクト指向データを上記対応するメタデータ・オブジェクトに関連付けられた対応するデータ・オブジェクトに配置する、インポート手段と、

上記オブジェクト指向コンピューティング環境内のオブジェクトから上記外部コンピューティング環境へオブジェクト指向データをエクスポートするための該オブジェクト指向コンピューティング環境からの要求に回答して、該オブジェクトに対応するメタデータ・オブジェクトを識別し、該メタデータ・オブジェクトに対応する上記外部データ名を上記データリストから識別し、対応するメタデータ・オブジェクト内の変換パラメータを上記オブジェクトからのオブジェクト指向データに適用し、それにより該オブジェクト指向データを外部データに変換するエクスポート手段と、

を含むデータ転送システム。

【請求項2】 1つまたは複数のコンピュータ・プラットフォーム上で動作する外部コンピューティング環境と、  
1つまたは複数のコンピューティング・プラットフォーム上で動作するオブジェクト指向コンピューティング環境との間でデータをインポート及びエクスポートするためのシステムであって、

2

上記オブジェクト指向コンピューティング環境が、外部コンピューティング環境データ・フィールド名と、対応するオブジェクト指向コンピューティング環境属性と、  
上記外部コンピューティング環境における各外部データ・フィールドに対応する外部データ形式とをリストするデータリスト・オブジェクトを含み、

上記オブジェクト指向コンピューティング環境が、対応する上記オブジェクト指向コンピューティング環境属性の1つを含む、上記オブジェクト指向コンピューティング環境内の各オブジェクトに対応するメタデータ・オブジェクトであって、対応する上記オブジェクト指向コンピューティング環境属性を上記対応する外部データ形式に変換するための変換パラメータを含むメタデータ・オブジェクトと、

外部データを上記オブジェクト指向コンピューティング環境にインポートするための上記外部コンピューティング環境からの要求に回答して、上記データリスト・オブジェクトからオブジェクト指向コンピューティング環境属性を識別し、対応するメタデータ・オブジェクトを識別し、該対応するメタデータ・オブジェクト内の変換パラメータを上記外部データに適用し、それにより該外部データをオブジェクト指向データに変換し、該変換されたオブジェクト指向データを上記対応するメタデータ・オブジェクトに関連付けられた対応するデータ・オブジェクトに配置するインポート手段と、

上記オブジェクト指向コンピューティング環境内のオブジェクトから上記外部コンピューティング環境へオブジェクト指向データをエクスポートするための該オブジェクト指向コンピューティング環境からの要求に回答して、該オブジェクトに対応するメタデータ・オブジェクトを識別し、該メタデータ・オブジェクトに対応する上記外部データ名を上記データリストから識別し、対応するメタデータ・オブジェクト内の変換パラメータを上記オブジェクトからのオブジェクト指向データに適用し、それにより該オブジェクト指向データを外部データに変換するエクスポート手段と、

を含むデータ転送システム。

【請求項3】 データをインポート及びエクスポートし、且つオブジェクトを見出すための外部オブジェクト・クラス・メソッドを含む外部オブジェクト・クラスと、  
対応する上記オブジェクト指向コンピューティング環境属性の1つを含む上記オブジェクト指向コンピューティング環境内の各オブジェクトを、上記外部オブジェクト・クラスから継承させ、それにより上記外部オブジェクト・クラス・メソッドを継承させる手段と、  
を含む、請求項1または2記載のシステム。

【請求項4】 対応する上記オブジェクト指向コンピューティング環境属性の1つを含む上記オブジェクト指向コンピューティング環境内の各オブジェクトにタグを追加する手段を含む、請求項1または2記載のシステム。

【請求項5】上記エクスポート手段が、上記変換された外部データを出力データ・ファイルに配置する手段を含む、請求項1または2記載のシステム。

【請求項6】1つまたは複数のコンピュータ・プラットフォーム上で動作する外部コンピューティング環境と、1つまたは複数のコンピューティング・プラットフォーム上で動作するオブジェクト指向コンピューティング環境との間でデータをインポート及びエクスポートするためのシステムであって、

データがエクスポートされインポートされる上記オブジェクト指向コンピューティング環境内の各オブジェクトを識別する手段と、

上記の各識別オブジェクトに対応するメタデータ・オブジェクトを生成するメタデータ生成手段であって、該メタデータ・オブジェクトが各識別オブジェクトの属性に対応する外部データ形式に変換する変換パラメータを含むメタデータ生成手段と、

外部データを上記オブジェクト指向コンピューティング環境にインポートするための上記外部コンピューティング環境からの要求に回答して、対応するメタデータ・オブジェクトを識別し、該対応するメタデータ・オブジェクト内の変換パラメータを上記外部データに適用し、それにより該外部データをオブジェクト指向データに変換し、該変換されたオブジェクト指向データを上記対応するメタデータ・オブジェクトに関連付けられた対応するデータ・オブジェクトに配置するインポート手段と、  
上記オブジェクト指向コンピューティング環境内のオブジェクトから上記外部コンピューティング環境へオブジェクト指向データをエクスポートするための該オブジェクト指向コンピューティング環境からの要求に回答して、該オブジェクトに対応するメタデータ・オブジェクトを識別し、該対応するメタデータ・オブジェクト内の変換パラメータを上記オブジェクト指向データに適用し、それにより該オブジェクト指向データを外部データに変換するエクスポート手段と、

を含むデータ転送システム。

【請求項7】データをインポート及びエクスポートし、且つオブジェクトを見い出すための外部オブジェクト・クラス・メソッドを含む外部オブジェクト・クラスと、上記の各識別オブジェクトを上記外部オブジェクト・クラスから継承させ、それにより上記外部オブジェクト・クラス・メソッドを継承させる手段と、  
を含む、請求項5記載のシステム。

【請求項8】上記の各識別オブジェクトにタグを追加する手段を含む、請求項5記載のシステム。

【請求項9】上記エクスポート手段が、上記変換された外部データを出力データ・ファイルに配置する手段を含む、請求項5記載のシステム。

【請求項10】1つまたは複数のコンピュータ・プラットフォーム上で動作する外部コンピューティング環境

と、1つまたは複数のコンピューティング・プラットフォーム上で動作するオブジェクト指向コンピューティング環境との間でデータをインポート及びエクスポートするためのデータ転送方法であって、

外部コンピューティング環境データ・フィールド名と、対応するオブジェクト指向コンピューティング環境属性と、外部コンピューティング環境における各外部データ・フィールドに対応する外部データ形式とをリストするデータリストを生成するステップと、

上記対応するオブジェクト指向コンピューティング環境属性の1つを含む、上記オブジェクト指向コンピューティング環境内の各オブジェクトに対応するメタデータ・オブジェクトを生成するステップと、

上記メタデータ・オブジェクトが、対応するオブジェクト指向コンピューティング環境属性を上記対応する外部データ形式に変換する変換パラメータを含み、該パラメータを適用することによってインポート及びエクスポートされるデータを変換するステップと、を具備するデータ転送方法。

【請求項11】データをインポート及びエクスポートし、且つオブジェクトを見つけ出すための外部オブジェクト・クラス・メソッドを含む外部オブジェクト・クラスを生成するステップと、

対応する上記オブジェクト指向コンピューティング環境属性の1つを含む上記オブジェクト指向コンピューティング環境内の各オブジェクトを、上記外部オブジェクト・クラスから継承させ、それにより上記外部オブジェクト・クラス・メソッドを継承させるステップと、

を含む、請求項10記載の方法。  
【請求項12】対応する上記オブジェクト指向コンピューティング環境属性の1つを含む上記オブジェクト指向コンピューティング環境内の各オブジェクトにタグを追加するステップを含む、請求項10記載の方法

【請求項13】上記データを上記オブジェクト指向コンピューティング環境にインポートするための上記外部コンピューティング環境からの要求に回答して、上記データリストからオブジェクト指向コンピューティング環境属性を識別するステップと、

対応するメタデータ・オブジェクトを識別するステップと、

対応する上記メタデータ・オブジェクト内の変換パラメータを上記データに適用し、それにより外部データをオブジェクト指向データに変換するステップと、

上記オブジェクト指向データを対応する上記メタデータ・オブジェクトに関連付けられた対応するデータ・オブジェクトに配置するステップと、

を含む、請求項10記載の方法。

【請求項14】上記オブジェクト指向コンピューティング環境内のオブジェクトから上記外部コンピューティング環境へデータをエクスポートするための該オブジェク

ト指向コンピューティング環境からの要求に应答して、該オブジェクトに対応するメタデータ・オブジェクトを識別するステップと、

上記メタデータ・オブジェクトに対応する外部データ名を上記データリストから識別するステップと、

対応する上記メタデータ・オブジェクト内の変換パラメータを上記オブジェクトからの上記データに適用し、それによりオブジェクト指向データを外部データに変換するステップと、

を含む、請求項10記載の方法。

【請求項15】上記データを出力データ・ファイルに配置するステップを含む、請求項14記載の方法。

【請求項16】1つまたは複数のコンピューティング・プラットフォーム上で動作する外部コンピューティング環境から、1つまたは複数のコンピュータ・プラットフォーム上で動作するオブジェクト指向コンピューティング環境ヘデータをインポートする方法であって、

上記オブジェクト指向コンピューティング環境が、外部コンピューティング環境データ・フィールド名と、対応するオブジェクト指向コンピューティング環境属性と、  
上記外部コンピューティング環境における各外部データ・フィールドに対応する外部データ形式とをリストするデータリスト・オブジェクトを含み、

上記オブジェクト指向コンピューティング環境が、対応する上記オブジェクト指向コンピューティング環境属性の1つを含む、上記オブジェクト指向コンピューティング環境内の各オブジェクトに対応するメタデータ・オブジェクトであって、

対応する上記オブジェクト指向コンピューティング環境属性を対応する上記外部データ形式に変換するための変換パラメータを含むメタデータ・オブジェクトを含み、  
上記データを上記オブジェクト指向コンピューティング環境ヘインポートするための上記外部コンピューティング環境からの要求に应答して、上記データリスト・オブジェクトからオブジェクト指向コンピューティング環境属性を識別するステップと、

対応するメタデータ・オブジェクトを識別するステップと、

上記対応するメタデータ・オブジェクト内の変換パラメータを上記外部データに適用し、それにより該外部データをオブジェクト指向データに変換するステップと、

上記変換されたオブジェクト指向データを対応する上記メタデータ・オブジェクトに関連付けられた対応するデータ・オブジェクト内に配置するステップと、

を含むデータ転送方法。

【請求項17】1つまたは複数のコンピューティング・プラットフォーム上で動作するオブジェクト指向コンピューティング環境から、1つまたは複数のコンピュータ・プラットフォーム上で動作する外部コンピューティング環境ヘデータをエクスポートする方法であって、

上記オブジェクト指向コンピューティング環境が、外部コンピューティング環境データ・フィールド名と、対応するオブジェクト指向コンピューティング環境属性と、  
上記外部コンピューティング環境における各外部データ・フィールドに対応する外部データ形式をリストするデータリスト・オブジェクトを含み、

上記オブジェクト指向コンピューティング環境が、対応する上記オブジェクト指向コンピューティング環境属性の1つを含む、上記オブジェクト指向コンピューティング環境内の各オブジェクトに対応するメタデータ・オブジェクトであって、対応する上記オブジェクト指向コンピューティング環境属性を対応する上記外部データ形式に変換するための変換パラメータを含むメタデータ・オブジェクトを含み、

上記オブジェクト指向コンピューティング環境内のオブジェクトから上記外部コンピューティング環境ヘデータをエクスポートするための該オブジェクト指向コンピューティング環境からの要求に应答して、該オブジェクトに対応するメタデータ・オブジェクトを識別するステップと、

上記メタデータ・オブジェクトに対応する外部データ名を上記データリストから識別するステップと、

対応するメタデータ・オブジェクト内の変換パラメータを上記オブジェクトからの上記データに適用し、それによりオブジェクト指向データを外部データに変換するステップと、

を含むデータ転送。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明はデータ処理システム及び方法に関し、特にオブジェクト指向のコンピューティング環境に関する。

【0002】

【従来の技術】オブジェクト指向のプログラミング・システム及び処理は“オブジェクト指向コンピューティング環境”とも称され、最新のデータ処理環境における研究及び関心の対象である。当業者には既知のように、オブジェクト指向コンピューティング環境は非常に多数の“オブジェクト”から構成される。オブジェクトは、“フレーム”とも呼ばれるデータ構造、並びにこのデータ構造をアクセス可能な“メソッド”とも呼ばれる動作または機能のセットである。フレームは多くの“スロット”を有し、それらの各々はスロット内のデータの“属性”を有する。属性はプリミティブ（整数またはストリングなど）であったり、オブジェクト参照（OREF）、すなわち別のオブジェクトに対するポインタであったりする。

【0003】各定義オブジェクトは、通常、多数の“インスタンス”において表現される。各インスタンスは特定のオブジェクト例に対応する特定のデータ構造を含

む。オブジェクト指向コンピューティング環境では、データはオブジェクトにそのメソッドの1つを実行するように要求することにより処理される。これはオブジェクトに"メッセージ"を送信することにより達成される。受信オブジェクトは、メッセージ名をインプリメントするメソッドを選択することによりメッセージに応答し、命名されたインスタンス上でこのメソッドを実行し、制御をメソッドの結果と共に呼出し側の高レベル・ルーチンに戻す。

【0004】オブジェクト指向コンピューティング環境が急増するにつれ、オブジェクト指向コンピューティング環境と外部コンピューティング環境との間のデータのインポート (import: 入力) 及びエクスポート (export: 出力) を許可することが、ますます重要となっている。外部コンピューティング環境は、従来の機能的にプログラムされたコンピューティング環境であったり、別のオブジェクト指向コンピューティング環境であったりする。オブジェクト指向環境と外部環境との間の典型的なインタフェースとしては、対話式ホスト・ユーザ・インタフェース、バッチ・インタフェース、遠隔ワークステーション・ユーザ・インタフェース、またはコンピューティング環境間の他の既知のインタフェースが含まれる。

【0005】オブジェクト指向コンピューティング環境と外部コンピューティング環境との間のデータのインポート及びエクスポートにおいて、インタフェースは、通常は、オブジェクト指向コンピューティング環境と外部コンピューティング環境との間で定義及び生成される。このインタフェースは、インポート・データ及びエクスポート・データの妥当性を検証するための、及びインポート・データ及びエクスポート・データをインポート・システムまたはエクスポート・システムにより使用される形式に変換するための機構を含む。インタフェースの数が急増するにつれ、データのインポート及びエクスポート機構は、各インタフェース型ごとに複製され、個別化されなければならない。個別にユーザ・インタフェースを生成し維持する必要があるため、プログラムの複雑化及びコストが増え、エラーの可能性も増加する。

【0006】

【発明が解決しようとする課題】本発明の目的は、このような従来技術の問題を踏まえて、改良されたオブジェクト指向コンピューティング環境を提供することである。

【0007】本発明の別の目的は、他のコンピューティング環境との間でデータをインポート及びエクスポートするためのシステム及び方法を含む、オブジェクト指向コンピューティング環境を提供することである。

【0008】更に本発明の別の目的は、オブジェクト指向環境内のオブジェクトに対する共通外部インタフェースを提供することである。

【0009】

【課題を解決するための手段】これらの目的が本発明によれば、オブジェクトに対する妥当性検証、変換及び更新制御機構を、オブジェクトに関する"メタデータ" (metadata) にカプセル化することにより達成される。このカプセル化は、メタデータ論理が様々な内部オブジェクトに対し同様に動作することを可能とし、各ユーザ・インタフェースに対応する同一の制御論理を複製する必要性から、オブジェクト指向コンピューティング環境を解放する。データのインポート及びエクスポートのための変換及び検証パラメータを定義するために、オブジェクトの属性を変更するタグ付け機構 (tagging scheme) を使用することにより、データがインポート及びエクスポートされる形式及びコンテキストが一般的に定義される。従って、オブジェクト指向コンピューティング環境と各外部コンピューティング環境との間で通信するための別のインタフェースは要求されない。インポートされる属性に対応して特殊な変換技術または妥当性検証が要求される時に限り、オブジェクトへの入力を変換及び検証する論理が要求される。

【0010】特に、外部コンピューティング環境とオブジェクト指向コンピューティング環境との間のデータのインポート及びエクスポートのためのシステムが、データリスト生成手段を含む。これはプログラム生成の間 (ビルド・タイム (build time)) に動作し、外部コンピューティング環境データ・フィールド名、対応するオブジェクト指向コンピューティング環境内部属性名、及び外部コンピューティング環境における各外部データ・フィールドに対応する外部データ形式をリストするデータリスト・オブジェクトを生成する。この外部データ形式は、オブジェクト指向コンピューティング環境と外部コンピューティング環境との間でインポート及びエクスポートされる。外部オブジェクト・クラスが定義され、これには対応するオブジェクト指向コンピューティング環境属性に対応する外部データ形式に変換するための変換パラメータが含まれる。外部オブジェクト・クラス・フレームは"メタデータ"とも称され、インポート及びエクスポートのために使用されるオブジェクト指向コンピューティング環境属性を含む。メタデータ生成手段はまた、プログラム生成時に、対応するオブジェクト指向コンピューティング環境属性の1つを含むオブジェクト指向環境内の各オブジェクトに対応して、メタデータ・オブジェクトを外部オブジェクト・クラスのサブクラスとして生成するために提供される。

【0011】プログラムの実行時 (ラン・タイム)、インポート手段は外部コンピューティング環境からの要求に応答して、外部データをオブジェクト指向コンピューティング環境にインポートする。インポート手段はデータリスト・オブジェクトを生成し、データリスト・オブジェクトからオブジェクト指向コンピューティング環境

の属性を識別し、対応するメタデータ・オブジェクトを識別する。インポート手段は対応するメタデータ・オブジェクト内の変換パラメータを外部データに適用し、それにより外部データをオブジェクト指向データに変換する。インポート手段はオブジェクト指向データを、対応するメタデータ・オブジェクトに関連する対応するデータ・オブジェクトに配置する。

【0012】プログラム実行時、エクスポート手段はオブジェクト指向環境からの要求に回答し、オブジェクト指向コンピューティング環境内のオブジェクトから外部コンピューティング環境にデータをエクスポートする。エクスポート手段はデータリストからメタデータ・オブジェクトに対応する外部データ名を読み出し、データリストから内部名を識別する。対応するメタデータ・オブジェクトが次に識別される。エクスポート手段は対応するメタデータ・オブジェクト内の変換パラメータをオブジェクトからのデータに適用し、それによりオブジェクト指向データを外部データに変換する。外部データは、外部コンピューティング環境により要求される特定の形式にフォーマットされる。

【0013】プログラム生成の間、本発明による方法は上述のメタデータ・オブジェクトを生成し、それによりオブジェクト指向コンピューティング環境が、実行時にインポート及びエクスポート要求に回答可能となる。実行中、データはオブジェクト指向システムにインポートされる。これは外部名及び値を読み出し、データリストから属性を識別するデータリスト・オブジェクトを生成し、対応するメタデータ・オブジェクトを識別することによって実施される。そして、メタデータ・オブジェクト内の変換パラメータを識別データに適用することにより、外部データをオブジェクト指向データに変換する。オブジェクト指向データは、メタデータ・オブジェクトに関連するデータ・オブジェクト内に配置される。

【0014】外部環境にデータをエクスポートするため、外部名がデータリストから読み出され、内部名がデータリストから識別される。対応するメタデータ・オブジェクトが識別される。メタデータ・オブジェクト内の変換パラメータがオブジェクトからのデータに適用され、オブジェクト指向データを外部データに変換する。従って、変換及び検証のための規則及び方法がオブジェクト指向コンピューティング環境に組込まれ、それにより、オブジェクト指向環境と通信する新たな外部環境に対応して、規則及び方法が生成される必要はない。新たな外部環境とのインタフェースは、最小の新プログラミング生成及び最小のエラー確率により、容易に追加及び変更される。

【0015】

【実施例】本発明の実施例を表す図を参照しながら、本発明を詳細に説明する。しかしながら、本発明は多くの異なる形式により実施され、本明細書において説明され

る実施例に限定されるものではない。これらの実施例は、本開示を明らかにするために提供され、本発明の範囲を当業者に伝達するものである。説明全体に渡り、同一番号が同一要素を参照するために使用される。

【0016】本発明によるデータのインポート（入力）及びエクスポート（出力）のためのシステム及び方法を説明する前に、オブジェクト指向コンピューティング環境の一般的な概要について説明する。それに続いて、本発明のシステム及び方法の一般的な説明をした後、本発明によるデータのインポート及びエクスポートのためのシステム及び方法に関する詳細な設計の説明を行う。

【0017】オブジェクト指向コンピューティング環境：オブジェクト指向コンピューティング環境では、作業はアクション要求メッセージをデータを含むオブジェクトに送信することにより実行される。オブジェクトは予め定義されるメソッドに従い、要求されるアクションをデータに対して実行する。オブジェクトはオブジェクト・クラスにグループ化され、オブジェクト・クラスはデータの型及び意味、及びクラス内のオブジェクトが尊重するアクション要求（メッセージ）を定義する。データを含む個々のオブジェクトはオブジェクトの“インスタンス”と称される。オブジェクト・クラスは他のクラスのサブクラスとして定義される。サブクラスは親クラスの全てのデータ特性及びメソッドを継承する（inherit）。これらはデータ及びメソッドを追加でき、親クラスのデータ要素またはメソッドをオーバーライドまたは再定義することができる。

【0018】オブジェクトまたはオブジェクト・クラスは、通常はリングによって表され、リングの中央にはスロットを有するフレーム（データ構造）が含まれる（図2、図3、図4参照）。各スロットはスロット内のデータの属性を含む。このリングのセクタはオブジェクトのメソッドを示すために使用される。このメソッドはフレームをカプセル化し、オブジェクト・インスタンスのフレーム内にカプセル化されたデータに対してアクションを実行する。

【0019】図1を参照すると、本発明がオペレートするハードウェア環境及びソフトウェア環境が示される。図2に示されるように、本発明は、1個または複数のコンピュータ・プラットフォーム12上で動作する外部コンピューティング環境11と、1個または複数のコンピュータ・プラットフォーム14上で動作するオブジェクト指向コンピューティング環境13との間のデータのインポート及びエクスポートのための方法及びシステムに関する。当業者には理解されるように、コンピュータ・プラットフォーム12及び14の各々は、通常、中央処理ユニット（CPU）17、主メモリ18、及び入出力（I/O）インタフェース19などのコンピュータ・ハードウェア・ユニット16を含み、更に表示端末21、キーボードまたはマウスなどの入力装置22、磁気ディ

スクまたは光ディスクなどの不揮発性データ記憶装置23、プリンタ24、及び他の周辺装置など周辺コンポーネントを含んでも良い。コンピュータ・プラットフォーム12または14は、通常、更にマイクロ命令コード26、及びオペレーティング・システム28を含む。

【0020】図1に示されるように、オブジェクト指向コンピューティング環境13はコンピュータ・プラットフォーム14上で動作し、外部コンピューティング環境11はコンピュータ・プラットフォーム12上で動作する。しかしながら、当業者には理解されるように、オブジェクト指向コンピューティング環境13及び外部コンピューティング環境11は、各々、複数のコンピュータ・プラットフォームを横断して動作したりする。それとは別に、オブジェクト指向コンピューティング環境13及び外部コンピューティング環境11が、同一のコンピュータ・プラットフォーム上で動作してもよい。更に、外部コンピュータ環境が、表示端末21及び入力装置22を含むワーク・ステーションの1部であるコンピュータ・プラットフォーム上で動作してもよい。

【0021】1例としては、各コンピュータ・プラットフォーム12及び14を、IBM社のシステム370アーキテクチャを有するコンピュータとすることが出来る。オペレーティング・システム28はIBM多重仮想記憶(MVS)オペレーティング・システムである。オブジェクト指向コンピューティング環境13は好適には中間C言語により書かれる。これはCコンピュータ・プログラミング言語のオブジェクト指向言語であり、C++コンピュータ・プログラミング言語に類似する。コンピュータ・プラットフォーム及びオブジェクト指向コンピューティング環境の設計及び動作は、当業者には既知であり、例えば、Abrahamらによる米国特許第5161225号"Persistent Stream for Processing Time Consuming and Reusable Queries in an Object Oriented Database Management System"(特願平2-230751号)、Abrahamらによる同第5151987号"Recover Objects in an Object Oriented Computing Environment"(特願平3-273553号)、及びAbrahamらによる同第5161223号"Resumeable Batch Query for Processing Time Consuming Queries in an Object Oriented Database Management System"(特願平2-254006号)、更にBertrand Meyerによる"Object Oriented Software Construction"(1988年、Prentice Hall発行)などの多くの文献で述べられている。

【0022】インポート及びエクスポート・システム：全体設計及び動作

図2を参照すると、本発明によるデータのインポート及びエクスポートのためのシステムが示される。図示のように、システムはオブジェクト指向コンピュータ環境13(図1)に含まれ、オブジェクト指向コンピューティング環境13と外部コンピューティング環境11との間

のデータのインポート及びエクスポートを提供する。図2に示されるように、インポート及びエクスポート・システムはデータリスト生成手段31を含み、これは外部コンピューティング環境11内のデータ・フィールド名、オブジェクト指向コンピューティング環境13内の対応する属性、及び外部コンピューティング環境11内の各外部データ・フィールドに対応する外部データ形式をリストするデータリスト・オブジェクト32を生成する。データリスト・オブジェクト32の詳細な設計及び構造については後述する。

【0023】図2を更に参照すると、インポート及びエクスポート手段はメタデータ生成手段33を含む。このメタデータ生成手段は、オブジェクト指向コンピューティング環境13内の各データ・オブジェクト36に対応するメタデータ・オブジェクト43を生成し、これにはデータリスト32からの対応するオブジェクト指向コンピューティング環境属性の1つが含まれる。以降で述べられるように、メタデータ・オブジェクト43は外部オブジェクト・クラスの属性であり、その属性及びメソッドの1部として、対応するオブジェクト指向コンピューティング環境属性に対応する外部データ形式に変換するためのパラメータ及びルーチンを含む。

【0024】図2を更に参照すると、インポート手段37は外部コンピューティング環境11からの要求に回答し、外部データをオブジェクト指向コンピューティング環境内にインポートする。インポート手段37はデータリスト・オブジェクト32からオブジェクト指向コンピューティング環境属性を識別し、対応するメタデータ・オブジェクト43を識別する。インポート手段37はまた、対応するメタデータ・オブジェクト43内の変換パラメータを外部コンピューティング環境11により提供されるデータに適用し、それによって外部データをオブジェクト指向データに変換し、このオブジェクト指向データを対応するメタデータ・オブジェクト43に関連して対応するデータ・オブジェクト36内に配置する。

【0025】図2を更に参照すると、インポート及びエクスポート・システムはエクスポート手段38を含み、これはオブジェクト指向コンピューティング環境からの要求に回答して、オブジェクト36から外部コンピューティング環境11にデータをエクスポートする。エクスポート手段38はオブジェクト36に対応するメタデータ・オブジェクト43を識別する。エクスポート手段38はまた、データリスト32からメタデータ・オブジェクト43に対応する外部データ名を識別する。最後に、エクスポート手段38は対応するメタデータ・オブジェクト43内の変換パラメータをオブジェクト36からのデータに適用し、それによってオブジェクト指向データを外部データに変換する。外部データは次に外部コンピュータ環境11に、既知の技術を使用することにより転送される。



【0026】図3を参照すると、データリスト・オブジェクト・クラス32の構造が示される。図示のように、データリスト・クラス・フレーム41は、外部名、内部名、及び値に関連する属性を含む。外部名は外部コンピューティング環境11内のデータ・フィールドの名前である。内部名は関連するデータ・オブジェクト36の属性の名前である。値は関連する外部名におけるデータの形式である。データリストはデータリスト・クラス・メソッド42a乃至42iを含み、これらは消去、追加、移動、設定、獲得、移動及び設定、生成、マッピング、及び返却にそれぞれ対応し、以降で詳細に説明される。

【0027】図4及び図5を参照すると、外部オブジェクト・クラス34の構造が示される。図示のように、外部オブジェクト・クラス34はメタデータ・フレーム43及び複数のメタデータ・メソッド44a乃至44hを含み、これらについては詳細に後述される。メタデータ・フレーム43はそれ自身メタデータ・オブジェクトを形成し、メタデータ属性46a乃至46nを含み、これらについても後述される。

【0028】図6を参照すると、インポート手段37及びエクスポート手段38（図2）はオブジェクト指向コンピューティング環境の生成の間（これを“ビルド・タイム”（build time）と呼ぶ）に動作する（ブロック51）。ブロック52で示されるように、インポートまたはエクスポートする全てのオブジェクト36は外部オブジェクト・クラス34から継承するように配列され、外部オブジェクト・クラスのメソッド44及び外部オブジェクト・クラスに関連するメタデータ・オブジェクト43がこれら全てのオブジェクトに適用される。

【0029】次にブロック53で、オブジェクト・タグがインポートまたはエクスポートする全てのオブジェクト36に追加され、以降で述べられるようにメタデータ・オブジェクトの生成に使用される。ブロック54の動作は、通常は、オブジェクト指向コンピューティング環境13上でオペレートするプログラマにより実行されるが、他のシステムがこれらのアクションを開始することも可能である。最後に、ブロック54で示されるように、メタデータ・オブジェクト43がインポートまたはエクスポートする各オブジェクト36に対応して生成される。メタデータ・オブジェクトは、以降で詳細に述べられるように、インポート及びエクスポート用のパラメータを定義する。

【0030】図7を参照すると、インポート手段37（図2）は、オブジェクト指向環境の実行中（“ラン・タイム”（Run Time）とも称される）の外部コンピューティング環境要求に回答して、データをインポートするように動作する。ブロック62では、外部名及び値がファイル、パネル、ウィンドウ、または外部コンピューティング環境11により提供される他のインポート機構から読出される。インポート手段は次にデータリスト・オ

ブジェクト32を生成する（ブロック63）。インポート手段37は次に適切なデータリスト32を使用し、対応するオブジェクトを識別する（ブロック64）。ブロック65では、識別されたオブジェクトに関連する対応するメタデータ・オブジェクトが識別される。次にブロック66で、識別されたメタデータ・オブジェクトを使用し、インポート型、インポート・メソッド、データ型、及び変換のために使用される他のパラメータが識別される。ブロック67では、識別されたパラメータにもとづき変換が実行される。最後にブロック68で、変換されたデータが識別されたオブジェクトに配置される。

【0031】図8を参照すると、エクスポート手段38（図2）はオブジェクト指向環境の実行中（“ラン・タイム”とも称される）、オブジェクト指向コンピューティング環境要求に回答して動作する（ブロック71）。ブロック72では、外部名がデータリストから読出され、ブロック73では内部名がデータリスト32から識別される。対応するメタデータ・オブジェクト43が次にブロック74で識別される。ブロック75では、メタデータ・オブジェクトから、エクスポート型、エクスポート・メソッド、データ型、及び変換用の他のパラメータが識別される。次にブロック76で変換が実行され、ブロック77で変換結果が出力ファイルに書込まれる。出力ファイルは次にコンピュータ・プラットフォーム14により、他の出力ファイルと同様に処理される。

【0032】インポート及びエクスポート・システム：詳細な設計及び動作

本発明によるインポート及びエクスポートのためのシステム及び方法の設計及び動作の概要を説明してきたが、次にこのシステム及び方法の設計及び動作について詳細に説明する。最初にデータリスト・オブジェクト・クラス32（図3）について、詳細な例をもとに説明する。次にインポート及びエクスポート・システム及び方法をインプリメントするために、オブジェクト・クラス及び属性に追加されるタグについて説明する。その次に、メタデータ・オブジェクト43を含む外部オブジェクト・クラス34について詳細に説明する。詳細な説明に続き、ビルド・タイム及びラン・タイムにおける本発明の動作を表す例が提示される。

【0033】データリスト・クラス（DATA\_LIST CLASS）：再度、図3を参照すると、データリスト・クラス32は配列

・Name : STRING (n) ;

・Value : STRING (n) ;

を含むオブジェクト・クラスである。

【0034】Nameiはインスタンス属性の名前を含み、次に示すシンタックスを有する。

Name1, Name2, Name3, . . . NameN

ここで、Nameiは現クラスに対応する内部名である。

【0035】例えば、型がPersonである、2つの属性Mo

m及びDadを含むFamilyオブジェクトを仮定する。また、Personオブジェクトが3つの属性、すなわちLONG型（プリミティブ）のSoc\_Sec\_Num、Phone型のPhone\_Number、及びDate型のBirth\_Dateを有するものと仮定する。Phoneオブジェクトは3つのプリミティブ属性、すなわちArea\_Code、Phone\_Num、及びExtensionを有する。Dateオブジェクトは3つのプリミティブ・オブジェクト、すなわちDate\_Num、Calendar、及びFormatを有する。図9はFamilyオブジェクトを表すものである。図10はFamilyオブジェクトに対応するデータリスト・フレーム41を表す。値は"ファジー (fuzzy)"データをサポートするワイルド・カード文字を含んでもよい。

【0036】データリスト・オブジェクト32のメソッド42は次の通りである。

消去 (delete) メソッド42a:

- ・内部名を与えられたエントリを消去
- ・外部名を与えられたエントリを消去

追加 (add) メソッド42b:

- ・この内部名のエントリを追加（これが既に存在しない場合）
- ・この外部名のエントリを追加（これが既に存在しない場合）

移動 (move) メソッド42c:

- ・この内部名のエントリに移動
- ・この外部名のエントリに移動

設定 (set) メソッド42d:

- ・現エントリの外部名を設定
- ・現エントリの内部名を設定
- ・現エントリの値を設定

獲得 (get) メソッド42e:

- ・現エントリの外部名を獲得
- ・現エントリの内部名を獲得
- ・現エントリの値を獲得

移動及び設定 (move and set) メソッド42f:

- ・この外部名のエントリに移動し、内部名及び値を設定
- ・この外部名のエントリに移動し、内部名を設定
- ・この外部名のエントリに移動し、値を設定
- ・この内部名のエントリに移動し、外部名を設定
- ・この内部名のエントリに移動し、値を設定

生成 (build) メソッド42g:

- ・全ての外部名のリストを生成
- ・全ての内部名のリストを生成

マップ (map) ・メソッド42h:

- ・このvarclassを用いて外部名を内部名にマップ
- ・このvarclassを用いて内部名を外部名にマップ

返却 (return) メソッド42i:

- ・合併リストを返却
- ・交差リストを返却
- ・不等値リストを返却

【0037】図11に示されるデータリスト・フレーム

の実施例では、エントリは任意の順序で記憶されるが、索引テーブルは外部名及び内部名によりエントリをアクセスするように維持される。これは外部名及び内部名の両者によるエントリの2進探索を可能とする。

【0038】また実施例では、変数フレーム領域がデータリスト・エントリを生成するために使用される。換言すれば、フレームは変数領域内の任意の構造の配列の処理方法を認識する変数領域スロット・オブジェクトを含む。スロット・オブジェクトは、Abrahamらによる米国特許出願第425746号"Slot Objects for an Object Oriented Programming System"（特願平2-251671号）で述べられている。

【0039】また、インポートまたはエクスポートされる各クラスに対応する内部名に、または内部名から外部名をマップする（割当てる）機構が提供される。この機構はクラスの処理時に常に変換及び維持されるビルド・タイム・マッピングか、または要求時に2つの名前間で変換を行うラン・タイム・マッピングである。

【0040】タグ (TAGS): オブジェクト指向コンピューティング環境分野の当業者には理解されるように、タグは基本属性の変更子 (modifier) である。本発明によれば、クラス・タグ及び属性タグが、インポートまたはエクスポートされるオブジェクト及びオブジェクト・クラスに追加される。追加される2つのクラス・タグは次の通りである。

・DEFAULT\_IMPORT

このタグは、属性に対応してATTRIBUTE\_IMPORTタグが宣言されない時に使用される、デフォルト・インポート手法を指定する。詳細についてはATTRIBUTE\_IMPORTタグに関する下記の記載を参照されたい。

・DEFAULT\_EXPORT

このタグは、属性に対応してATTRIBUTE\_EXPORTタグが宣言されない時に使用される、デフォルト・エクスポート手法を指定する。詳細についてはATTRIBUTE\_IMPORTタグに関する下記の記載を参照されたい。

【0041】更に、3つの属性タグが追加される。属性タグは属性に続き、且つ属性の最初のコメント・ラインより前のラインの最初に現れる。望ましくは、継承される属性を再定義し、これらのタグを再指定しても良い。再指定は全ての子によっても同様に継承される。これらの属性タグは任意の属性と一緒に使用可能であり、次の通りである。

・UNIQUE\_KEY

このタグはこの属性をこのオブジェクトの特定のキーを構成する属性の1つとして識別する。すなわち、これらの属性はオブジェクトを一意的に識別する。

・ATTRIBUTE\_IMPORT

このタグは属性がインポートされる方法を指定する。これはこの属性に対応するDEFAULT\_IMPORTタグだけをオーバーライドする。ATTRIBUTE\_IMPORTタグが指定されない場

合、Import\_Data\_ListがDEFAULT\_IMPORTタグにより指定されるアクションをデフォルト指定とする。ATTRIBUTE\_IMPORTタグについては以降で詳細に述べる。

・ATTRIBUTE\_EXPORT

このタグは属性がエクスポートされる方法を指定する。これはこの属性に対応するDEFAULT\_EXPORTタグだけをオーバーライドする。ATTRIBUTE\_EXPORTタグが指定されない場合、Export\_Dataが DEFAULT\_EXPORTタグにより指定されるアクションをデフォルト指定とする。ATTRIBUTE\_EXPORTタグについては以降で詳細に述べる。

【0042】ATTRIBUTE\_IMPORT: ATTRIBUTE\_IMPORTタグのシンタクスは次の通りである。

--ATTRIBUTE\_IMPORT: "value"; ここで、"value"は次に示す内のいずれかである。

・IGNORE

この属性はインポートされない。

・ATTRIBUTE

この属性はインポートされる。Assign\_Data は、属性自身をメッセージ呼出しのオブジェクトとして使用することにより、属性の静的クラスのImport\_Data を呼出す。これは属性がこのようにして使用されること、すなわち、属性がオブジェクト参照またはスロット・オブジェクトのいずれかであることを意味する。インポートされる特定の属性がVIRTUAL 型の場合、この値は無効であり、シンタクス・エラーを生じる。それ以外の場合、つまりエクスポートされる特定の属性がオブジェクト参照またはスロット・オブジェクトまたはVIRTUALでない場合、Assign\_Data はデータの入力形式を属性のデータ型に変換し、これを直接フレームに記憶する。仮想属性 (VIRTUAL) は、フレーム内には割当てられる記憶域を有さないが、Meta\_Data オブジェクト内にはエントリを有する属性である。そのシンタクスは次の通りである。

attribute\_name: VIRTUAL;

・ATTRIBUTE alternate\_method

この属性はインポートされる。Assign\_Data は、属性自身をメッセージ呼出しのオブジェクトとして使用することにより、属性の静的クラスのalternate\_methodを呼出す。これは属性がこのようにして使用されること、すなわち、属性がオブジェクト参照またはスロット・オブジェクトのいずれかであることを意味する。インポートされる特定の属性がオブジェクト参照またはスロット・オブジェクトでない場合、この値は無効であり、シンタクス・エラーを生じる。alternate\_methodは、Import\_Data メソッドに対応して以前に確立された基準に従わねばならない。

・set\_attribute\_method

この属性はインポートされる。Assign\_Data は、自身をメッセージ呼出しのオブジェクトとして使用することにより、指定されるset\_attribute\_methodを呼出す。これは自身を含むクラスのメソッドである。このメソッドは

フィールド・レベル検証だけを実行するべきである。他の検証 (例えばクロス・フィールド及びクロス・オブジェクト検証) は、Am\_I\_Validメソッドに配置されるべきである。set\_attribute\_methodは次の呼出しインタフェースに従う。

```
set_attribute_method (
value_in: in "primitive";
rc_in_out: in out RETURN_CODE_CLASS);
```

サポートされるプリミティブのリストに対応する外部クラス内の Assign\_Dataメソッドを参照されたい。変数パラメータ・リストをサポートするCの機能により、上記呼出し形式は次に定義されるメソッドと共に作用する。

```
set_attribute_method (value_in: in "primitive");
```

【0043】ATTRIBUTE\_EXPORT: ATTRIBUTE\_EXPORTタグのシンタクスは次の通りである。

--ATTRIBUTE\_EXPORT: "value";

ここで"value"は次に示す内のいずれかである。

・EXCLUDE

この属性はエクスポートされない。

・ATTRIBUTE

この属性はエクスポートされる。属性がオブジェクト参照またはスロット・オブジェクトの場合、Export\_Data はその属性自身をメッセージ呼出しのオブジェクトとして使用することにより、その属性のExport\_Data メソッドを呼出す。エクスポートされる特定の属性がVIRTUAL型の場合、この値は無効であり、シンタクス・エラーを生じる。エクスポートされる特定の属性が、オブジェクト参照またはスロット・オブジェクトまたはVIRTUALでない場合、すなわちこれがプリミティブの場合、Export\_Data は属性データをフレームから直接取り出し、これを属性のデータ型からストリングに変換する。

・ATTRIBUTE KEYS\_ONLY

この属性はエクスポートされる。Export\_Data は、属性自身をメッセージ呼出しのオブジェクトとして使用することにより、属性のGet\_Keyメソッドを呼出す。これは属性がこのように使用されること、すなわち、属性がオブジェクト参照またはスロット・オブジェクトのいずれかであることを意味する。エクスポートされる特定の属性がオブジェクト参照またはスロット・オブジェクトでない場合、この値は無効であり、シンタクス・エラーを生じる。

・ATTRIBUTE alternate\_method

この属性はエクスポートされる。Export\_Data は、属性自身をメッセージ呼出しのオブジェクトとして使用することにより、その属性のExport\_Data メソッドを呼出す。これは属性がこのようにして使用されること、すなわち、属性がオブジェクト参照またはスロット・オブジェクトのいずれかであることを意味する。エクスポートされる特定の属性がオブジェクト参照またはスロット・オブジェクトでない場合、この値は無効であり、シンタ

クス・エラーを生じる。alternate\_methodは、Export\_Data メソッドに対応して以前に確立された基準に従わねばならない。どの属性がデータリストに追加されるかの判断は、alternate\_methodの責任である。

・get\_attribute\_method

この属性はエクスポートされる。Export\_Data は自身をメッセージ呼出しのオブジェクトとして使用することにより、指定されるget\_attribute\_methodを呼出す。これは自身を含むクラスのメソッドである。get\_attribute\_methodは次の呼出しインタフェースに従う。

get\_attribute\_method (

value\_out : out "primitive";

rc\_in\_out : in out RETURN\_CODE\_CLASS) ;

サポートされるプリミティブのリストに対応する外部クラス内の Assign\_Dataメソッドを参照されたい。変数パラメータ・リストをサポートするCの機能により、上記呼出し形式は次に定義されるメソッドと共に作用する。

get\_attribute\_method (

value\_out : out "primitive") ;

ほとんどの属性アクセス・メソッドが、これらサポートされる2つの形式の一方に既に相当するため、これは既存のコードに対する影響を最小化する。

【0044】外部オブジェクト・クラス (EXTERNAL OBJECT CLASS) : 外部オブジェクト・クラス34 (図4及び図5) は抽象親オブジェクト・クラスであり、これは内部オブジェクト指向コンピューティング環境13と外界コンピューティング環境11との間のデータ対話を提供する。外部クラスは次の機構を有する。

・Meta\_Dataフレーム、43

このクラス・フレームはクラスの機構に関する情報をカプセル化する。Meta\_Dataオブジェクトはビルド・タイムの間に生成及びポビュレートされる。

・Import\_Dataメソッド、44a

このクラス・メソッドは1個または複数のエントリのデータリストを受入れ、

1. Import\_Findメソッドを呼出し、次に、

2. Import\_Find メソッドにより返却されるインスタンスを使用することにより、Assign\_Data\_Listメソッドを呼出す。

・Get\_Keyメソッド、44b

このインスタンス・メソッドは全てのUNIQUE\_KEYフィールドの名前と値を含むデータリストを返却する。

・Object\_Searchメソッド、44c

このクラス・メソッドはデータリストを受取り、参照をデータリスト内で見い出されるキー値に一致するイン・メモリ・オブジェクトのインスタンスに返却する。

・Import\_Findメソッド、44d

このクラス・メソッドはデータリストを受取り、参照をデータリスト内で見い出されるキー値に一致するオブジェクトのインスタンスに返却する。

・Export\_Dataメソッド、44e

このインスタンス・メソッドはインスタンスに対応するデータリストを返却する。

・Assign\_Data\_Listメソッド、44f

このインスタンス・メソッドはデータリストを受取り、リスト内の各エントリを処理する。

・Assign\_Dataメソッド、44g

このインスタンス・メソッドはデータリスト名及び値を受取り、クラスのメタデータに依存して値を割当てて。

10 ・Am\_I\_Validメソッド、44h

このインスタンス・メソッドは、インスタンスが有効状態であることを保証するために、インスタンスをチェックする。例えば、このメソッドはクロス・フィールド及びクロス・オブジェクトの検証を実行する。

【0045】メタデータ・オブジェクト (META\_DATA OBJECT) 、43

このオブジェクトはクラスの機構に関する情報をカプセル化する。メタデータ・オブジェクトはビルド・タイムの間に生成及びポビュレートされる。これは包含するクラス内の各属性に関する次の情報を含む。

・内部属性名、46a

これは属性の名前である。

・内部データ型、46b

これは属性のデータ型 (すなわち静的クラス) である。

・クラス/インスタンス標識、46c

これは属性がクラス属性か、インスタンス属性かを示す。

・内部ロケーション、46d

30 これはエンクローズ (解放しない) するインスタンス・フレームまたはクラス・データ領域内の属性のロケーションである。

・インポート型、46e

これはAssign\_DataがATTRIBUTE\_IMPORT タグにより定義される属性を割当てて方法を示す。有効値は次の通りである。

- IGNORE

- ATTRIBUTE

- OVERRIDE

・インポート代替メソッド、46f

40 これはATTRIBUTE\_IMPORT : ATTRIBUTE\_alternate\_method が指定される時、デフォルトのImport\_Dataメソッドの代わりに呼出されるメソッドである。

・属性設定メソッド、46g

これはATTRIBUTE\_IMPORT : set\_attribute\_methodが指定される時、デフォルトのImport\_Dataメソッドの代わりに呼出されるメソッドである。

・属性設定メソッド・データ型、46h

これはset\_attribute\_methodの第1のパラメータとして渡される値のデータ型である。これはset\_attribute\_methodの第1のパラメータとしてコード化されるデータ型

21

を見ることにより獲得される。

・インポート最適化情報、46 i

インポートの間のメッセージの最適化を促進するための情報がここに記憶される。

・エクスポート型、46 j

これはエクスポート・データがATTRIBUTE\_EXPORTタグまたはDEFAULT\_EXPORTタグにより定義される属性をエクスポートする方法を示す。有効値は次の通りである。

- EXCLUDE
- ATTRIBUTE
- UNIQUE\_KEY
- ATTRIBUTE\_ALL
- OVERRIDE

・エクスポート代替メソッド、46 k

これはATTRIBUTE\_EXPORT: ATTRIBUTE\_alternate\_methodが指定される時、Export\_Dataメソッドの代わりに呼出されるメソッドである。

・属性獲得メソッド、46 l

これはATTRIBUTE\_EXPORT: get\_attribute\_methodが指定される時、エクスポートされる属性値を検索するために呼出されるメソッドである。

・属性獲得メソッド・データ型、46 m

これは第1のパラメータとしてget\_attribute\_methodに渡される値のデータ型である。これはget\_attribute\_methodの第1のパラメータとしてコード化されるデータ型を見ることにより獲得される。

・エクスポート最適化情報、46 n

エクスポートの間のメッセージの最適化を促進するための情報がここに記憶される。

【0046】IMPORT\_DATAメソッド、44 a: このクラス・メソッドは1個または複数のエントリのデータリストを受入れ、

1. Import\_Findメソッドを呼出し、次に、
2. Import\_Findメソッドにより返却されるインスタンスを使用することにより、Assign\_Data\_Listメソッドを呼出す。Import\_Dataメソッドとのインタフェースは次の通りである。

```
import_data (
data_list_in_out: in out DATA_LIST_CLASS;
rc_in_out      : in out RETURN_CODE_CLASS);
```

【0047】GET\_KEYメソッド、44 b:

このインスタンス・メソッドは全てのUNIQUE\_KEYフィールドの名前と値を含むデータリストを返却する。Get\_Keyメソッドとのインタフェースは次の通りである。

```
get_key (
data_list_out: out DATA_LIST_CLASS;
rc_in_out    : in out RETURN_CODE_CLASS);
```

【0048】OBJECT\_SEARCHメソッド、44 c: このクラス・メソッドはデータリストを受取り、データリスト内で見い出されるキー値に一致するイン・メモリ・オブ

22

ジェクトのインスタンスに参照を返却する。Object\_Searchメソッドとのインタフェースは次の通りである。

```
object_search (
data_list_in: in DATA_LIST_CLASS;
rc_in_out   : in out RETURN_CODE_CLASS)
return OREF;
```

このメソッドの振舞いは次の通りである。

1. 現作業単位 (Unit Of Work) UOW レベルで開始して、UOWレベル0に至るまで、このUOWインスタンス内のオブジェクトを探索する。これはこのクラスのUOWインスタンス内の各オブジェクトのGet\_Keyメソッドを呼出し、それをdata\_list\_inと比較することにより行われる。一致が見い出されるとオブジェクトのオブジェクト参照 (OREF) を返却する。作業単位については、Abraham らによる1989年10月23日出願の米国特許出願第425607号"Unit of Work for Preserving Data Integrity of a Database"で述べられている。
2. このクラスのグローバルUOWインスタンス内の各オブジェクトのGet\_Keyメソッドを呼出し、それをdata\_list\_inと比較することにより、グローバルUOWインスタンス内のオブジェクトを探索する。一致が見い出されるとオブジェクトのOREFを返却する。
3. 一致が見い出されない場合、VOID\_OREF を返却する。オブジェクトがメモリ内に見い出されない場合、データベース探索を追加するために、子がこのメソッドの再定義を希望することが予想される。

【0049】IMPORT\_FINDメソッド、44 d: このクラス・メソッドはデータリストを受取り、参照をデータリスト内で見い出されるキー値に一致するオブジェクトのインスタンスに返却する。Import\_Findメソッドとのインタフェースは次の通りである。

```
import_find (
data_list_in_out: in out DATA_LIST_CLASS;
rc_in_out       : in out RETURN_CODE_CLASS)
return OREF;
```

デフォルトの振舞いに関し、インスタンスがデータベース内に存在しない場合に、新たなインスタンスが生成されるものと仮定する。この仮定をサポートするために、クラス属性Import\_Objが、新たなインスタンスの生成を最適化するために使用される。効果的な論理フローは次の通りである。

1. Object\_Findを呼出し、非ボイド (non-void) のOREFを返却する。
2. それ以外の場合、Import\_Obj. VOID = TRUEならば、SELF.unflat (Import\_Objのアドレス)。
3. rc\_in\_outをOKにセットする。
4. Import\_Obj. copyを返却する。

【0050】EXPORT\_DATAメソッド、44 e: このインスタンス・メソッドはインスタンスに対応するデータリ

ストを返却する。データリストが既に内部名を有する場合、データリスト内に既に存在するそれらの名前に対応する値だけが記入される。データリストが空 (empty) の場合、名前及び値を含むエントリが全てのエクスポート可能な属性に対応してデータリストに追加される。Export\_Data メソッドとのインタフェースは次の通りである。

```
export_data (
data_list_in_out : in out DATA_LIST_CLASS ;
rc_in_out       : in out RETURN_CODE_CLASS) ;
```

【0051】 ASSIGN\_DATA\_LISTメソッド、44f: このインスタンス・メソッドはデータリストを受取り、

1. IGNOREエントリを"除去"する (すなわち新たなフィルタされたデータリストにnon-IGNOREエントリをコピーする)。
2. インスタンスに対応するデータリストを獲得するために、SELF. Export\_Dataを呼出す。
3. このデータリストをフィルタされたデータリストと比較し、一致する名前及び値を有するエントリをフィルタされたデータリストから除去する。

注釈: これは全ての不要な割当てを除去するわけではない。しかしながら、それらのほとんどを除去するべきである。次に示す可能性がある。

- a. インポート値がエクスポートされない。
- b. インポート値がエクスポート値と実際に異なる形式である (例えば、パーソナルIDが入力され、人のフル・ネームが出力される)。

4. フィルタリングされるデータリスト内の各内部名サブセットに対応するAssign\_Data\_Listを呼出す。またはフィルタされるデータリスト内の各単一の内部名に対応するAssign\_Dataを呼出す。
5. データリストの処理により起こされる状態遷移が有効であることを保証するために、Am\_I\_Validメソッドを呼出す。Assign\_Data\_Listメソッドとのインタフェースは次の通りである。

```
assign_data_list (
data_list_in_out : in out DATA_LIST_CLASS ;
rc_in_out       : in out RETURN_CODE_CLASS) ;
```

【0052】 ASSIGN\_DATAメソッド、44g: このインスタンス・メソッドはデータリストを受取り、クラスのメタデータに依存して、以下の処理のいずれかを実行する。

- ・データをプリミティブに変換し、それをインスタンスに詰め込む。

- ・次の"name, "タグを使用し、属性の静的クラスのメタデータの処理を継続する。
- ・属性の割当て方法を判断させるために、含まれるオブジェクトのAssign\_Data\_Listメソッドに処理をカスケードする。
- ・データをプリミティブに変換し、適切なインスタンス sel\_attribute\_methodを呼出す。Assign\_Dataメソッドとのインタフェースは次の通りである。

```
assign_data (
10 data_list_in_out : in out DATA_LIST_CLASS ;
rc_in_out       : in out RETURN_CODE_CLASS) ;
```

サポートされるプリミティブは次の通りである。

- ・ BOOLEAN
- ・ CHAR
- ・ DB2\_VARCHAR
- ・ DOUBLE
- ・ LONG
- ・ REAL
- ・ SHORT
- 20 ・ STRING
- ・ NLS\_DOUBLE
- ・ NLS\_LONG
- ・ NLS\_REAL
- ・ NLS\_SHORT
- ・ NLS\_STRING

【0053】 ビルド・タイム例: 図6に一般に示されるようなプログラム生成の間 (ビルド・タイム) におけるインポート・データ手段及びエクスポート・データ手段の動作を説明するために、5つのビルド・タイム例が提供される。次のテーブルでは、属性設定メソッド・データ型が設定メソッド・データ型と省略され、属性獲得メソッド・データ型が獲得メソッド・データ型と省略される。これらの例では、次に示す内容がクラス・プロログにおいてコード化されるものと仮定する。

```
package body MY_CLASS は、
...
--DEFAULT_IMPORT : ATTRIBUTE ;
【0054】 例1)
```

```
attr_1 : LONG ;
```

40 --\* この属性はインポートにおいても、エクスポートにおいても、特殊な処理を必要としない。デフォルトのインポート及びエクスポート特性が適用される。この例の属性に対応するメタデータを表1に示す。

【表1】

表1. メタデータ例1						
内部名	内部データ型	内部ローケーション	インポート型	インポート代替メソッド	設定メソッド	設定メソッドの戻り型
attr_1	LONG	0xF123				
インポート最適化情報	エクスポート型	インポート代替メソッド	強引メソッド	強引メソッドの戻り型	インポート最適化情報	

【0055】例2)

attr\_2: LONG;

-- ATTRIBUTE\_IMPORT: imp\_attr\_2;

-- ATTRIBUTE\_EXPORT: exp\_attr\_2;

--\* この属性はインポートにおいて特殊な処理を要し、メソッドはこのクラス内のimp\_attr\_2メソッドである。

--\* この属性はエクスポートにおいて特殊な処理を要し、メソッドはこのクラス内のexp\_attr\_2メソッドである。ここで、imp\_attr\_2メソッドは次のように定義され\*

10\*る。

imp\_attr\_2 (

value\_in: in LONG) ;

また、exp\_attr\_2は次のように定義される。

exp\_attr\_2 (

value\_out: out LONG) ;

この属性例に対応するメタデータを表2に表す。

【表2】

表2. メタデータ例2						
内部名	内部データ型	内部ローケーション	インポート型	インポート代替メソッド	設定メソッド	設定メソッドの戻り型
attr_2	LONG	0xF456				
インポート最適化情報	エクスポート型	インポート代替メソッド	強引メソッド	強引メソッドの戻り型	インポート最適化情報	

【0056】例3)

attr\_3: MY\_SLOT;

-- ATTRIBUTE\_IMPORT: put\_name;

-- ATTRIBUTE\_EXPORT: get\_name;

--\* この属性はインポートにおいて特殊な処理を要し、メソッドはこのクラス内のput\_nameメソッドである。

--\* この属性はエクスポートにおいて特殊な処理を要し、メソッドはこのクラス内のget\_nameメソッドである。ここで、put\_nameは次のように定義される。

※put\_name (

30 value\_in : in DB2\_VARCHAR;

rc\_in\_out: in out RETURN\_CODE\_CLASS) ;

また、get\_nameは次のように定義される。

get\_name (

value\_out: out DB2\_VARCHAR;

rc\_in\_out: in out RETURN\_CODE\_CLASS) ;

この例のメタデータを表3に表す。

※ 【表3】

表3. メタデータ例3						
内部名	内部データ型	内部ローケーション	インポート型	インポート代替メソッド	設定メソッド	設定メソッドの戻り型
attr_3	MY_SLOT	0xF789				
インポート最適化情報	エクスポート型	インポート代替メソッド	強引メソッド	強引メソッドの戻り型	インポート最適化情報	

【0057】例4)

attr\_4: MY\_FRAME;

--\* DEFAULT\_IMPORT設定を使用。

--\* この属性はインポートにおいて特殊な処理を要し、

メソッドはMY\_FRAMEクラス内に見い出される。

--\* DEFAULT\_EXPORT設定を使用。

--\* この属性はエクスポートにおいて特殊な処理を要し、メソッドはMY\_FRAMEクラス内に見い出される。こ

で、MY\_FRAMEはCLASS\_IMPORT: findを指定し、findメソッドは次のように定義されるクラス・メソッドである。

```
find (
value_in: in STRING;
rc_in_out: in out RETURN_CODE_CLASS)
returns MY_FRAME;
```

また、MY\_FRAMEはCLASS\_EXPORT: unfindを指定し、unfi \*

\*ndメソッドは次のように定義されるインスタンス・メソッドである。

```
unfind (
value_out: out STRING;
rc_in_out: in out RETURN_CODE_CLASS);
```

この属性例に対応するメタデータを表4に表す。

【表4】

表4. メタデータ例4						
内部名	内部データ型	内部ローケーション	インポート型	インポート代替メソッド	設定メソッド	設定メソッドの戻り型
attr_4	MY_FRAME	0x123				
インポート最適化情報	エクスポート型	インポート代替メソッド	設定メソッド	戻りメソッドの戻り型	インポート最適化情報	

【0058】例5)

attr\_5: MY\_SLOT;

--\* DEFAULT\_IMPORT設定を使用。

--\* この属性はインポートにおいて特殊な処理を要し、メソッドは MY\_SLOTクラス内に見い出される。

--\* DEFAULT\_EXPORT設定を使用。

--\* この属性はエクスポートにおいて特殊な処理を要し、メソッドはMY\_SLOTクラス内に見い出される。ここで、MY\_SLOTはCLASS\_IMPORT: assign\_str;を指定し、assign\_str は次のように定義されるクラス・メソッドである。

※assign\_str (

value\_in : in STRING;

rc\_in\_out : in out RETURN\_CODE\_CLASS;

また、MY\_SLOTはCLASS\_EXPORT: get\_str;を指定し、get\_str は次のように定義されるインスタンス・メソッドである。

get\_str (

value\_in\_out: in out STRING);

この属性例に対応するメタデータを表5に表す。

【表5】

※

表5. メタデータ例5						
内部名	内部データ型	内部ローケーション	インポート型	インポート代替メソッド	設定メソッド	設定メソッドの戻り型
attr_5	MY_SLOT					
インポート最適化情報	エクスポート型	インポート代替メソッド	設定メソッド	戻りメソッドの戻り型	インポート最適化情報	

【0059】ラン・タイム例：一般的に図7及び図8に示されるような、プログラム実行の間（ラン・タイム）のデータのインポート及びエクスポートを説明するために、ラン・タイムにおけるインポート手段及びエクスポート手段の動作例について示す。下記に示される擬似コード例は擬似コード・ファイルのフラグメントであり、2つの別の領域、すなわちデータ辞書及びそのオブジェクトに対応するオブジェクト・データを含む。2つの領域は2つのブランク・ラインにより分離される。

【0060】インポート：インポートの間、最終オブジェクト検証が post\_unflatメソッドにおいて実行される。インポート・メソッドは典型的には、割当てられる属性に関連して割当てられるデータの実際の値だけを検証する。しかしながら、全ての個々のフィールドが別々

に有効であるが、オブジェクト自身が無効状態である状況が存在する。こうした場合は、オブジェクトの"Am\_1.Valid"メソッドが Post\_Unflatメソッドから呼出される。また、インポート手段37がその外部名をメタデータ内に見い出すことのできない属性を読込む場合、これはDEFAULT\_IMPORTメソッドを呼出し、その属性及びその値を受渡す。

【0061】次に示す例は、前章の5つの"ビルド・タイム例"に対応する。

【0062】例1) インポート手段37がインポート手段37ファイル内に、次に示す内容を見い出すものと仮定する。

DEFID, 4, EKXMYCLS; attr\_1, LONG; . . .

4; (1): 123; . . . . .



以下の処理が発生する。

1. インポート手段37がデータリストから属性の内部名"attr\_1"、及び属性の外部値"123"を読込む。
2. インポート手段37が内部名"attr\_1"に対応してインポートされるクラスのメタデータを調査し、それを見出す。
3. インポート手段37が次に属性のインポート型を見て、ATTRIBUTEを見出す。
4. インポート手段37が次に属性の内部データ型を見て、LONGを見出す。
5. インポート手段37が"123"を(LONG)123に変換する。
6. インポート手段37が属性の内部オフセットを見て、0xF123を見出す。
7. インポート手段37が(LONG)123をフレームのオフセット0xF123に挿入する。

【0063】例2) インポート手段37がファイル内に次に示す内容を見出すものと仮定する。

```
DEFID, 4, EKNMYCLS; attr_2, LONG; . . .
4; (2): 123; . . . .
```

以下の処理が発生する。

1. インポート手段37が属性の内部名"attr\_2"、及び属性の外部値"456"を読込む。
2. インポート手段37が内部名"attr\_2"に対応してインポートされるクラスのメタデータを調査し、それを見出す。
3. インポート手段37が次に属性のインポート型を見て、SELFを見出す。
4. インポート手段37が次に属性のインポート・メソッドを見て、imp\_attr\_2を見出す。
5. インポート手段37が次にインポート・メソッドのデータ型を見て、LONGを見出す。
6. インポート手段37が"123"を(LONG)123に変換する。
7. インポート手段37がローカルRCをno-error、no-message状態に初期化する。
8. インポート手段37がインポートされるオブジェクトのimp\_attr\_2を呼出し、(LONG)123を第1パラメータとして受渡す。
9. インポート手段37がエラー状態に対応するRCパラメータを評価する。

【0064】例3) インポート手段37がファイル内に次に示す内容を見出すものと仮定する。

```
attr_3, 4, EKNMYCLS; Company Name, MY_SLOT; . . .
4; (2): 'Acme Business Machines'; . . .
```

以下の処理が発生する。

1. インポート手段37が属性の内部名"attr\_3"、及び属性の外部値"AcmeBusiness Machines"を読込む。
2. インポート手段37が内部名"attr\_3"に対応してインポートされるクラスのメタデータを調査し、それを見

い出す。

3. インポート手段37が次に属性のインポート型を見て、SELFを見出す。
4. インポート手段37が次に属性のインポート・メソッドを見て、pul\_nameを見出す。
5. インポート手段37が次にインポート・メソッドのデータ型を見て、VARCHARを見出す。
6. インポート手段37が"Acme Business Machines"をvarcharに変換する。
7. インポート手段37がローカルRCをno-error、no-message状態に初期化する。
8. インポート手段37がインポートされるオブジェクトのpul\_nameを呼出し、varcharを第1パラメータとして受渡す。
9. インポート手段37がエラー状態に対応するRCパラメータを評価する。

【0065】例4) インポート手段37がファイル内に次に示す内容を見出すものと仮定する。

```
DEFID, 4, EKNMYCLS; attr_4, MY_FRAME; . . .
4; (2): 'Acme Business Machines'; . . .
```

この例は外部名の使用が正しくない場合を仮定する。いずれの場合にも、MY\_FRAMEを選択される名前に変更する。以下の処理が発生する。

1. インポート手段37が属性の内部名"attr\_4"、及び属性の外部値"AcmeBusiness Machines"を読込む。
2. インポート手段37が内部名"attr\_4"に対応してインポートされるクラスのメタデータを調査し、それを見出す。
3. インポート手段37が次に属性のインポート型を見て、ATTRIBUTEを見出す。
4. インポート手段37が次に属性のデータ型を見て、MY\_FRAMEを見出す。
5. インポート手段37が次にインポート・キー・メソッドに対応するMY\_FRAMEに対応するメタクラス・オブジェクトを見て、findを見出す。
6. インポート手段37が次にインポート・キー・メソッドのデータ型に対応するMY\_FRAMEに対応するメタクラス・オブジェクトを見て、STRINGを見出す。
7. インポート手段37がローカルRCをno-error、no-message状態に初期化する。

8. インポート手段37がメッセージattr\_4.findを送信し、STRINGを第1パラメータとして受渡す。
9. インポート手段37がエラー状態に対応するRCパラメータを評価する。

【0066】例5) インポート手段37がファイル内に次に示す内容を見出すものと仮定する。

```
DEFID, 4, EKNMYCLS; attr_5 string, MY_SLOT, 6; . . .
```

```
DEFID, 6, EKNMYSLT; attr_5 string, STRING; . . .
```

```
4; (2): 'Acme Business Machines'; . . .
```

以下の処理が発生する。

1. インポート手段37が属性の内部名"attr\_5"、及び属性の外部値"AcmeBusiness Machines"を読み込む。
  2. インポート手段37が内部名"attr\_5"に対応してインポートされるクラスのメタデータを調査し、それを見出す。
  3. インポート手段37が次に属性のインポート型を見て、ATTRIBUTEを見出す。
  4. インポート手段37が次に属性のデータ型を見て、MY\_SLOTを見出す。
  5. インポート手段37が次にインポート・キー・メソッドに対応するMY\_SLOTに対応するメタクラス・オブジェクトを見て、assign\_strを見出す。
  6. インポート手段37が次にインポート・キー・メソッドのデータ型に対応するMY\_SLOTに対応するメタクラス・オブジェクトを見て、STRINGを見出す。
  7. インポート手段37がローカルRCをno-error、no-message状態に初期化する。
  8. インポート手段37がメッセージattr\_5. assign\_strを送信し、STRINGを第1パラメータとして受渡す。
  9. インポート手段37がエラー状態に対応するRCパラメータを評価する。
- 【0067】エクスポート：各インポート例が次にエクスポート・モードにおいて説明され、これらはエクスポート手段38により実行される。
- 【0068】例1) この例はattr\_1のエクスポートに関連するステップを説明する。
1. エクスポート手段38がエクスポートされるクラスのメタデータを調査し、内部名attr\_1を見出す。
  2. エクスポート手段38が次に外部名を探し、"attr\_1"を見出す。
  3. エクスポート手段38が外部名を出力ファイルに書込む。
  4. エクスポート手段38が属性のエクスポート型を探し、ATTRIBUTEを見出す。
  5. エクスポート手段38が次に属性の内部データ型を探し、LONGを見出す。
  6. エクスポート手段38が囲むフレーム内の属性のオフセットを探し、0xF123を見出す。
  7. エクスポート手段38がオフセット0xF123のLONGをSTRINGに変換し、それを出力ファイルに書込む。出力は次の通りである。
- ```
DEFID, 4, EKNMYCLS; attr_1, LONG; . . . .
4; (1): 123; . . . .
```
- 【0069】例2) この例は、attr\_2のエクスポートに関連するステップを調査する。
1. エクスポート手段38がエクスポートされるクラスのメタデータを調査し、内部名attr\_2を見出す。
  2. エクスポート手段38が次に外部名を探し、"Long

Value"を見出す。

3. エクスポート手段38が外部名を出力ファイルに書込む。
  4. エクスポート手段38が属性のエクスポート型を探し、SELFを見出す。
  5. エクスポート手段38が次にエクスポート・メソッドを探し、exp\_attr\_2を見出す。
  6. エクスポート手段38がエクスポート・メソッド・データ型を探し、LONGを見出す。
  7. エクスポート手段38がローカルRCをno-error、no-messages状態に初期化する。
  8. エクスポート手段38がエクスポートされるインスタンスのexp\_attr\_2を呼出し、LONGのアドレスを第1パラメータとして受渡す。
  9. エクスポート手段38がエラー状態に対応するRCパラメータを評価する。
  10. エクスポート手段38がLONGをSTRINGに変換し、それを出力ファイルに書込む。出力は次の通りである。
- ```
DEFID, 4, EKNMYCLS; attr_2, LONG; . . . .
4; (2): 123; . . . .
```
- 【0070】例3) この例はattr\_3のエクスポートに関連するステップを調査する。
1. エクスポート手段38がエクスポートされるクラスのメタデータを調査し、内部名attr\_3を見出す。
  2. エクスポート手段38が次に外部名を探し、"Company Name"を見出す。
  3. エクスポート手段38が外部名を出力ファイルに書込む。
  4. エクスポート手段38が属性のエクスポート型を探し、SELFを見出す。
  5. エクスポート手段38が次にエクスポート・メソッドを探し、get\_nameを見出す。
  6. エクスポート手段38がエクスポート・メソッド・データ型を探し、VARCHARを見出す。
  7. エクスポート手段38がローカルRCをno-error、no-messages状態に初期化する。
  8. エクスポート手段38がエクスポートされるインスタンスのget\_nameを呼出し、VARCHARのアドレスを第1パラメータとして受渡す。
  9. エクスポート手段38がエラー状態に対応するRCパラメータを評価する。
  10. エクスポート手段38がVARCHARをSTRINGに変換し、それを出力ファイルに書込む。出力は次の通りである。
- ```
DEFID, 4, EKNMYCLS; Company Name, STRING; . . . .
4; (2): 'Acme Business Machines'; . . . .
```
- 【0071】例4) この例は、attr\_4のエクスポートに関連するステップを調査する。
1. エクスポート手段38がエクスポートされるクラス

のメタデータを調査し、内部名attr\_4を見い出す。

2. エクスポート手段38が次に外部名を探し、"attr\_4"を見い出す。

3. エクスポート手段38が外部名を出力ファイルに書込む。

4. エクスポート手段38が属性のエクスポート型を探し、ATTRIBUTEを見い出す。

5. エクスポート手段38が次に属性のデータ型を見て、MY\_FRAMEを見い出す。

6. エクスポート手段38が次にエクスポート・キー・メソッドに対応するMY\_FRAMEに対応するメタクラス・オブジェクトを見て、unfindを見い出す。

7. エクスポート手段38が次にエクスポート・キー・メソッドのデータ型に対応するMY\_FRAMEに対応するメタクラス・オブジェクトを見て、STRINGを見い出す。

8. エクスポート手段38がローカルRCをno-error、no-messages 状態に初期化する。

9. エクスポート手段38がメッセージattr\_4. unfindを送信し、ストリングを第1パラメータとして受渡す。

10. エクスポート手段38がエラー状態に対応するRCパラメータを評価する。

11. エクスポート手段38が結果ストリングを出力ファイルに書込む。出力は次の通りである。

```
DEFID, 4, EKNMYCLS; attr_4, MY_FRAME; . . .
```

```
4; (2): 'Acme Business Machines'; . . .
```

この例は正しくない外部名の使用を仮定する。いずれの場合にも、MY\_FRAMEを我々が決定する名前に変更する。

【0072】例5) この例はattr\_5のエクスポートに関連するステップを調査する。

1. エクスポート手段38がエクスポートされるクラスのメタデータを調査し、内部名attr\_5を見い出す。

2. エクスポート手段38が次に外部名を探し、"NLS String"を見い出す。

3. エクスポート手段38が外部名を出力ファイルに書込む。

4. エクスポート手段38が属性のエクスポート型を探し、ATTRIBUTEを見い出す。

5. エクスポート手段38が次に属性のデータ型を見て、MY\_SLOTを見い出す。

6. エクスポート手段38が次にエクスポート・キー・メソッドに対応するMY\_SLOTに対応するメタクラス・オブジェクトを見て、get\_strを見い出す。

7. エクスポート手段38が次にエクスポート・キー・メソッドのデータ型に対応するMY\_SLOTに対応するメタクラス・オブジェクトを見て、STRINGを見い出す。

8. エクスポート手段38がローカルRCをno-error、no-messages 状態に初期化する。

9. エクスポート手段38がメッセージattr\_5. get\_strを送信し、ストリングを第1パラメータとして受渡す。

10. エクスポート手段38がエラー状態に対応するRCパラメータを評価する。

11. エクスポート手段38が結果ストリングを出力ファイルに書込む。出力は次の通りである。

```
DEFID, 4, EKNMYCLS; NLS String, SLOT, 6; . . .
```

```
DEFID, 6, EKNMYSLT; NLS String, STRING; . . .
```

```
4; (2): 'Acme Business Machines'; . . .
```

【0073】

【発明の効果】以上説明したように、本発明によれば、改良されたオブジェクト指向コンピューティング環境が提供され、外部のコンピューティング環境との間で、データをインポート及びエクスポートすることが可能となる。

【図面の簡単な説明】

【図1】オブジェクト指向コンピューティング環境、外部コンピューティング環境及びそれらに関連するハードウェア・プラットフォームを表すブロック図である。

【図2】本発明によりデータをインポート及びエクスポートするシステムのブロック図である。

【図3】図2のデータリスト・クラスの構造を表す図である。

【図4】図2の外部オブジェクト・クラスの構造を表す図である。

【図5】図2の外部オブジェクト・クラスの構造を表す図である。

【図6】プログラム生成時に図2のインポート手段及びエクスポート手段により実行される動作を表す図である。

【図7】プログラム生成時に図2のインポート手段により実行される動作を表す図である。

【図8】プログラム生成時に図2のエクスポート手段により実行される動作を表す図である。

【図9】オブジェクト例を表す図である。

【図10】図9のオブジェクト例のデータリスト・フレームを表す図である。

【図11】データリスト・フレームの実施例を表す図である。

【符号の説明】

11 外部コンピューティング環境

12、14 コンピュータ・プラットフォーム

13 オブジェクト指向コンピューティング環境

16 コンピュータ・ハードウェア・ユニット

21 表示端末

23 不揮発性データ記憶装置

26 マイクロ命令コード

31 データリスト生成手段

32 データリスト・オブジェクト

33 メタデータ生成手段

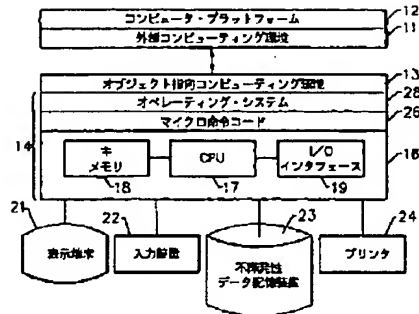
36 データ・オブジェクト

37 インポート手段

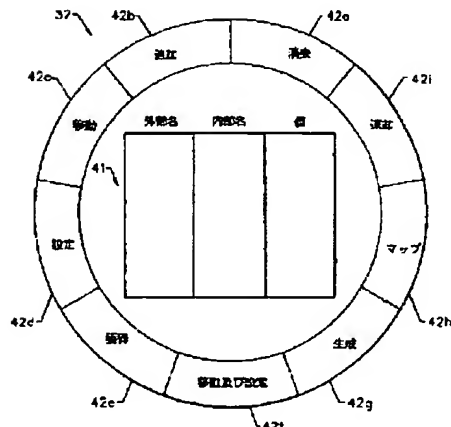
38 エクスポート手段

41 データリスト・クラス・フレーム

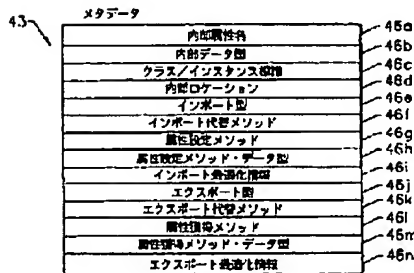
【図1】



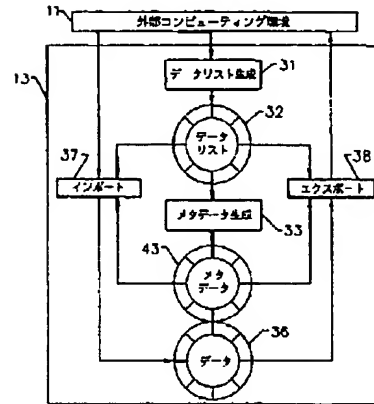
【図3】



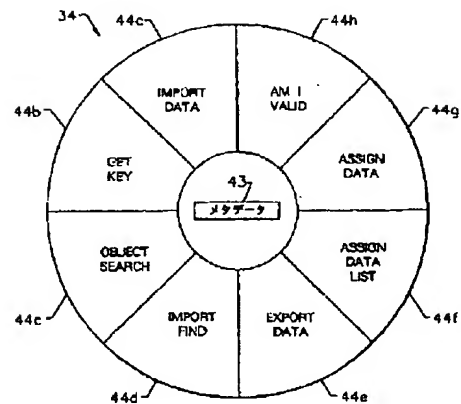
【図5】



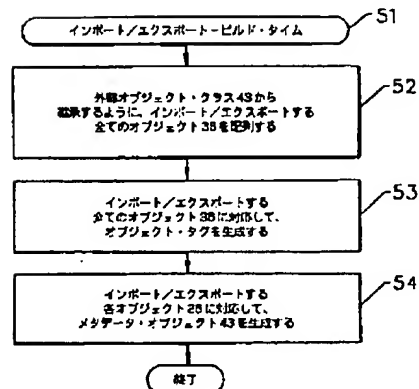
【図2】



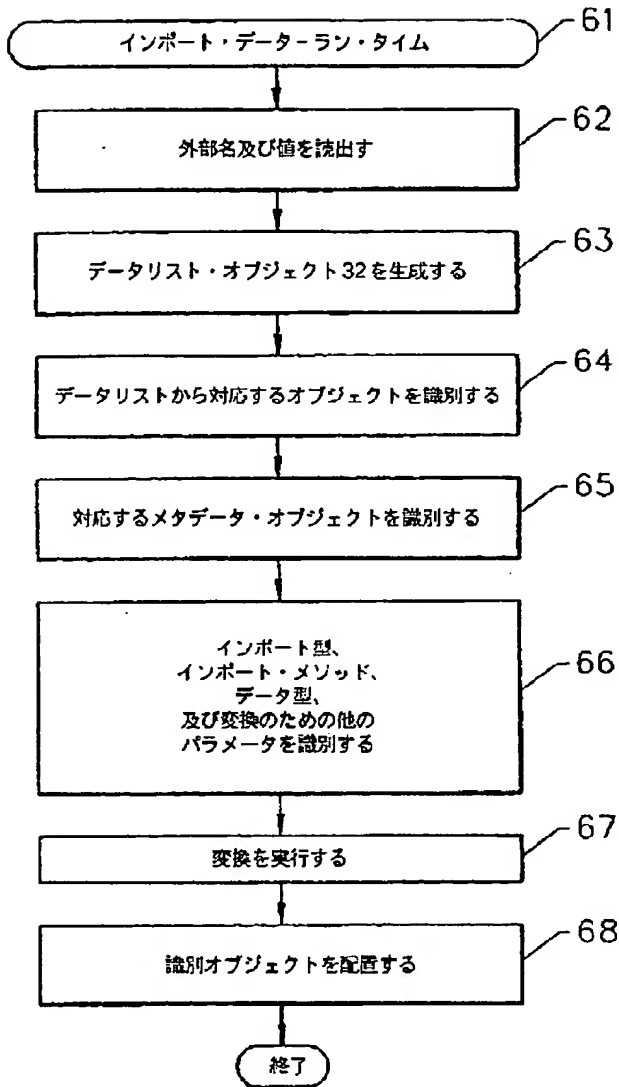
【図4】



【図6】



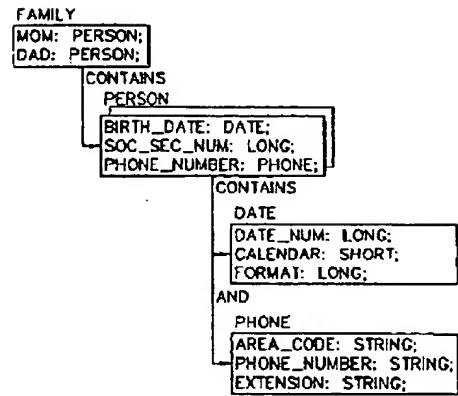
【図 7】



【図 10】

| 外部名                          | 内部名                        | 値         |
|------------------------------|----------------------------|-----------|
| MOM'S BIRTHDAY DATE NUM      | MOM.BIRTH_DATE.DATE_NUM    | 123456    |
| MOM'S BIRTHDAY CALENDAR      | MOM.BIRTH_DATE.CALENDAR    | 5         |
| MOM'S BIRTHDAY FORMAT        | MOM.BIRTH_DATE.FORMAT      | 654       |
| MOM'S SOCIAL SECURITY NUMBER | MOM.SOC_SEC_NUM            | 123456789 |
| MOM'S AREA CODE              | MOM.PHONE_NUMBER.AREA_CODE | 404       |
| MOM'S PHONE NUMBER           | MOM.PHONE_NUMBER.PHONE_NUM | 456-7890  |
| DAD'S BIRTHDAY DATE NUM      | DAD.BIRTH_DATE.DATE_NUM    | 234567    |
| DAD'S BIRTHDAY CALENDAR      | DAD.BIRTH_DATE.CALENDAR    | 5         |
| DAD'S BIRTHDAY FORMAT        | DAD.BIRTH_DATE.FORMAT      | 654       |
| DAD'S SOCIAL SECURITY NUMBER | DAD.SOC_SEC_NUM            | 564321876 |
| DAD'S AREA CODE              | DAD.PHONE_NUMBER.AREA_CODE | 404       |
| DAD'S PHONE NUMBER           | DAD.PHONE_NUMBER.PHONE_NUM | 678-2345  |
| DAD'S PHONE NUMBER           | DAD.PHONE_NUMBER.EXTENSION | 123       |

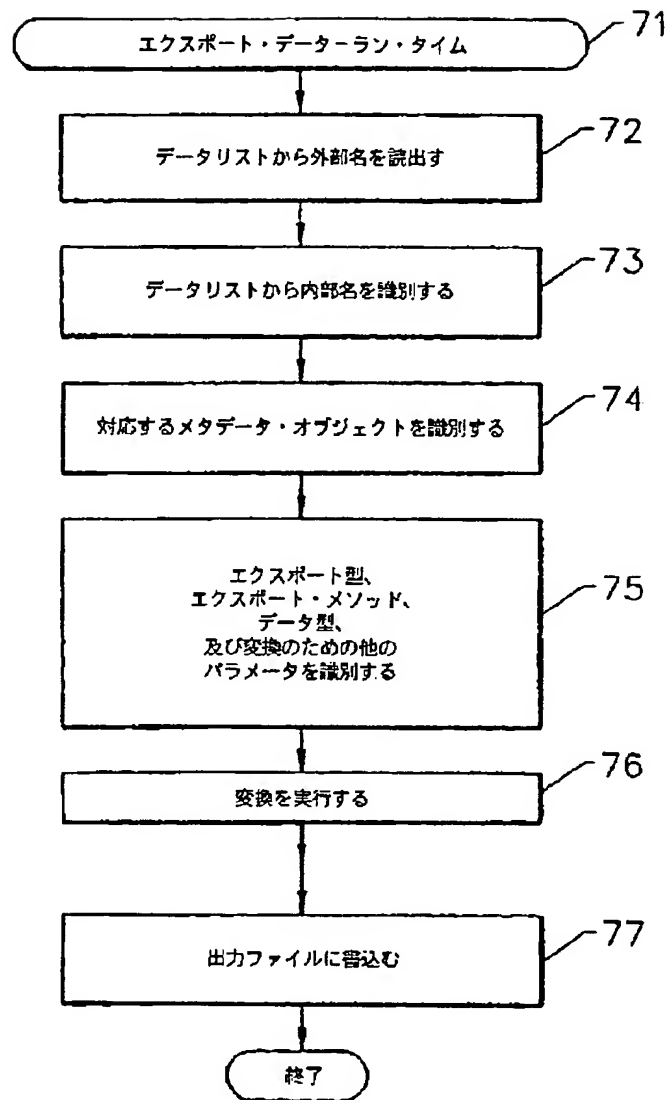
【図 9】



【図 11】

| データリスト・エントリ |    |       |        |     |           |
|-------------|----|-------|--------|-----|-----------|
| 外部名<br>番号   | 番号 | 外部名   | 内部名    | 値   | 内部名<br>番号 |
| 3           | 1  | FRED  | APPLE  | 123 | 1         |
| 1           | 2  | SALLY | PEAR   | 456 | 4         |
| 5           | 3  | BOB   | PEACH  | 789 | 5         |
| 2           | 4  | TED   | BANANA | 234 | 3         |
| 4           | 5  | JIMMY | KIWI   | 567 | 2         |

【図8】



フロントページの続き

(72)発明者 ウィリアム・エル・リッチ  
アメリカ合衆国30088、ジョージア州スト  
ーン・マウンテン、マーティン・ロード  
1112

(72)発明者 フロイド・ダブリュ・シャケルフォード  
アメリカ合衆国30518、ジョージア州ビュ  
ーフォード、ハノバー・ドライブ 3510

This Page Is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☒ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE (S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

### **IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**